Khayyati, S.* and **Tan, B.**, "A Lab-scale Manufacturing System Environment to Investigate Data-Driven Production Control Approaches," *Journal of Manufacturing Systems*, Vol.60, July, pp. 283-297, 2021.

https://doi.org/10.1016/j.jmsy.2021.06.002

©2021. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/

A Lab-scale Manufacturing System Environment to Investigate Data-Driven Production Control Approaches

Siamak Khayyati[a], Barış Tan[a,b]

[a] *College of Engineering, Koç University*
[b] *College of Administrative Sciences and Economics, Koç University*

Abstract

Controlling production and release of material into a manufacturing system effectively can lower work-in-progress inventory and cycle time while ensuring the desired throughput. With the extensive data collected from manufacturing systems, developing an effective real-time control policy helps achieving this goal. Validating new control methods using the real manufacturing systems may not be possible before implementation. Similarly, using simulation models can result in overlooking critical aspects of the performance of a new control method. In order to overcome these shortcomings, using a lab-scale physical model of a given manufacturing system can be beneficial. We discuss the construction and the usage of a lab-scale physical model to investigate the implementation of a data-driven production control policy in a production/inventory system. As a data-driven production control policy, the marking-dependent threshold policy is used. This policy leverages the partial information gathered from the demand and production processes by using joint simulation and optimization to determine the optimal thresholds. We illustrate the construction of the lab-scale model by using LEGO Technic parts and controlling the model with the marking-dependent policy with the data collected from the system. By collecting data directly from the lab-scale production/inventory system, we show how and why the analytical modeling of the system can be erroneous in predicting the dynamics of the system and how it can be improved. These errors affect optimization of the system using these models adversely. In comparison, the data-driven method presented in this study is considerably less prone to be affected by the differences between the physical system and its analytical representation. These experiments show that using a lab-scale manufacturing system environment is very useful to investigate different data-driven control policies before their implementation and the marking-dependent threshold policy is an effective data-driven policy to optimize material flow in manufacturing systems.

Keywords: Data-driven Control, Analysis of Manufacturing Systems, Simulation, Physical Models

Email addresses: skhayyati13@ku.edu.tr (Siamak Khayyati), btan@ku.edu.tr (Barış Tan)

1. Introduction

A modern manufacturing system has the necessary technological infrastructure to collect extensive data from the shopfloor and control all the processes. Utilizing this infrastructure requires developing effective data-driven production control methods. An important aspect of devising new control policies for manufacturing systems is validating them using a testbed. Testing new control policies in a real manufacturing system may not be possible before actual implementation or can be extremely costly. Furthermore, even if the implementation is possible in an actual manufacturing system, observing the long-run performance of the system under a proposed policy may take a very long time. As an alternative, simulation models are commonly used as digital twins for the validation of new policies. However, inadequacy of modeling the physical properties of the elements in a system in a sufficiently detailed way has been identified as one of the limitations of the simulation models. Simplifying assumptions used in the simulation models, ignoring smaller delays in the processes related to material handling, and human errors and testing a proposed policy with such a simulation model may yield an inaccurate evaluation of the proposed policy in a real implementation [1]. Using a design that is based on a simulation model that does not capture the physical properties of the system accurately can yield a lower efficiency of the system compared to target level [2].

Using a lab-scale physical manufacturing system model is a realistic middle ground alternative to using simulation models or actual manufacturing systems as testbeds. In many different areas, physical models have been used for performing tests to evaluate the performance, and finalizing and validating a given design. For example, in automotive industry, physical models are commonly used to finalize the design and evaluate the performance in more realistic settings [3]. Similarly, in construction industry, physical models of buildings are also used to visualize the design and performing tests [4]. In order to test the proposed optimization algorithm for a variant of the traveling salesman problem, drones and ground vehicles are also used as a testbed [5]. However, the use of physical models in design and control of manufacturing systems is limited.

Similar to the limitation of simulation models in other settings, validating control policies using lab-scale manufacturing system models includes many steps that are present in real systems but are usually ignored in simulation models. For example, transportation and handling of material can introduce considerable delays or unexpected failures may happen. Furthermore, using a lab-scale environment emphasizes the implementation issues such as considering the availability of the existing sensors, the communication system, data collection and computational capacity during the algorithm development phase. Therefore, lab-scale physical manufacturing system models can provide a testbed for studies that consider the processes of data gathering from a physical system, processing the collected data to build the models or estimate the system parameters, optimizing the system, and implementing the prescribed solutions in the physical system. As a result, lab-scale physical models can be used to develop more realistic digital twins and for demonstrating the feasibility of the proposed design [6, 7, 8, 9].

In this work, we discuss development of a lab-scale manufacturing system environment to investigate data-driven production control approaches. Developing data-driven methods is a part of the goals of Industry 4.0 and one of the necessities of smart manufacturing [10, 11]. Data-driven approaches have been used in different aspects of analysis of manufacturing systems including identifying critical processes in semiconductor manufacturing and bottleneck detection in serial lines [12, 13]. We focus on a specific policy that is referred as the marking-dependent production control policy implemented in a production/inventory system [14]. This policy has been devised in line with the criteria of data-driven methods that emphasise the direct use of data and avoiding imposing mathematical assumptions on the data for the purpose of tractability [15, 16]. Production/inventory systems have been extensively studied in the literature and the base-stock policy has been shown to be optimal for them, given restrictive assumptions such as independent arrivals, independent service times and a fully observable system [17]. For more complex settings such as deteriorating production and uncertain inventory, reinforcement learning and dynamic programming based methods have been proposed [18, 19]. For partially observable systems, variants of the base-stock policy have been studied [20, 21]. The marking-dependent threshold policy has been extended to more general systems by using a machine learning approach to implement these policies [22].

In the construction of the lab-scale model, we use LEGO Technic parts. The LEGO manufacturing system consists of conveyors, workstations, gates, sensors and EV3 bricks that enable the connection of the physical system with the computer that controls the setup. We introduce the different components of the setup and demonstrate the control of the system with the making-dependent threshold policy. Then, in order to validate the implementation of the data-driven control method, we conduct experiments where for given parameters for the setup, the actual inter-event times generated by the setup are collected and compared to that of the analytical models. Finally, we apply the data-driven joint simulation and optimization method to the traces gathered from the physical system.

In recent years, several LEGO manufacturing systems have been constructed and used mainly for educational purposes [23, 24, 25, 26]. Most of these implementations use the lab-scale models of manufacturing systems with reliable and unreliable stations, closed-loop and open-loop production lines. LEGO manufacturing systems have been used as a proof of concept for the digital twin and real-time simulation [27, 28, 29]. LEGO manufacturing systems have also been used for automatic generation of simulation models based on data collected from manufacturing systems without additional knowledge about the structure of the system [30]. Compared to the number of studies that report the use of lab-scale models for teaching purposes, the number of studies that report the use of these models for manufacturing system research is limited.

The main contributions of this study are two fold. First, we demonstrate the construction and the usage of a lab-scale manufacturing system model to investigate production control policies in detail with the relevant methods and algorithms. We discuss the disparity between the analytical models and the empirical data collected from the physical model and the ways to improve the models. Second, we validate the effectiveness of the data-driven marking-dependent production control policy in a production/inventory system by using a lab-scale manufacturing system model.

The remainder of this paper is organized as follows. Section 2 introduces the production/inventory system and the data-driven marking-dependent policy we use for validation. Section 3 describes the LEGO production/inventory system with construction of system elements and the centralized control algorithm. Section 4 discusses the results of the experiments conducted by using the lab-scale manufacturing system model. Finally, Section 5 concludes the paper.

2. Data-driven marking-dependent threshold policy

In the lab-scale manufacturing environment, a specific data-driven control policy that is referred as *the marking-dependent threshold policy* is used in a specific production/inventory system. The production/inventory model considered here falls under the category of make-to-stock queues that have been extensively studied in the literature. The state-dependent policies have been studied for these models under different conditions [31, 21]. For a survey of make-to-stock queues we refer the reader to [32].

In this section, the control policy and the production/inventory system are introduced following the setting given in [14]. Figure 1 shows a general setting where the real-time signals from the production time process and the demand process are collected and used by the control policy to decide whether the release of material into the production will be allowed. The marking-dependent threshold policy uses the clusters of real-time information signals, referred as *markings,* collected from a system to authorize the release of material into different parts of the system.

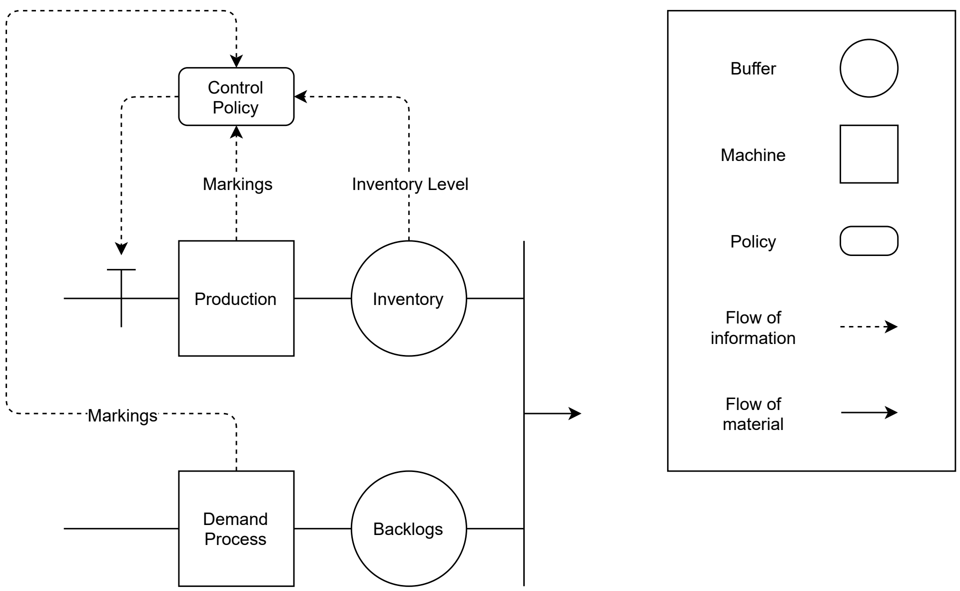


Figure 1: The marking-dependent threshold policy for a production/inventory system

The control problem considered in this study is minimizing the long run inventory and backlog costs by deciding whether the release of material into the production will be authorized based on the inventory level, the availability of the machine, and the last collected information signals related to the demand and production times that are represented by the selected markings. The formal definition of the policy, the model and its assumptions are given in the following part.

2.1. Model

We consider a discrete state space and continuous time representation of a manufacturing system. The state of the production time process at time t is $\eta_W(t) \in \{1, \dots, w\}$ where w

is the number of the different production time states. The working status of the production stage at time t is $M(t) \in \{0, 1\}$ where $M(t) = 0$ indicates that the production stage is idle and $M(t) = 1$ indicates that it is working. When the production starts, it cannot be preempted.

The state of the demand arrival process is $\eta_D(t) \in \{1, \ldots, d\}$, where d is the number of the different demand arrival states. In the experimental setup described in Section 2.5, the output from a two-station production line generates the demand for the production/inventory system, and the demand inter-arrival times are correlated.

The inventory position that is the difference between the cumulative production and cumulative demand at time t is $X(t)$, the inventory level is $X^+(t) = \max\{X(t), 0\}$ and the backlog level is $X^-(t) = \max\{-X(t), 0\}$. The inventory carrying cost is c^+ and the backlog cost is c^- per unit time for each part.

The markings arrive as the information signals with the demand arrivals or production completions. They can also arrive independently. The last observed marking from the information and demand process at time t is $c_D(t) \in \{1, \ldots, C_D\}$ where C_D is the number of markings for the demand process. The last observed marking from the production time process is $c_W(t) \in \{1, \ldots, C_W\}$ where C_W is the number of markings for the production process. The inventory status, the production status, and the marking processes, $(X(t), M(t), c_D(t), c_W(t))$ are observable. However, the demand and production states, denoted as $\eta_D(t)$ and $\eta_W(t)$ respectively, are not fully observable. The marking-dependent control policy is based on the observable state of the system.

2.2. Production Control Problem

The decision to authorize production at time t under policy l is denoted with

$$u^l(X(t), M(t), c_D(t), c_W(t)),$$

where $u^l = 1$ denotes authorizing the production and $u^l = 0$ denotes not authorizing the production depending on the inventory position, the machine status, and the last observed demand and production markings. The control problem is finding the optimal policy l that minimizes the steady-state average cost π in the long run

$$\pi^* = \min_l J^l = E\left[\frac{1}{T} \lim_{T\to\infty} \int_{t=0}^{T} \left(c^+X^+(t) + c^-X^-(t)\right) dt | X(0), \eta_D(0), \eta_W(0)\right]. \tag{1}$$

2.3. Marking-dependent Threshold Policy

The proposed solution to the control problem given in Equation (1) is using the marking-dependent threshold policy where the inventory position at time t is compared with the threshold level set for the last observed markings of the demand and production processes to authorize production. Let S_{c_D,c_W} be the threshold level for the marking pair (c_D, c_W), $c_D \in \{1, 2, ..., C_D\}$, $c_W \in \{1, 2, ..., C_W\}$. The arrival of a demand while the last observed marking pair is (c_D, c_W) authorizes production if there is no item being produced at that moment and the inventory level is less than or equal to S_{c_D,c_W}. Upon completion of a part, production is allowed to continue if the inventory level is less than S_{c_D,c_W}. If the inventory level reaches S_{c_D,c_W}, the production stops until a new demand or information signal arrives. The production will not start if the threshold that corresponds to the newly observed marking pair is lower than the inventory level. The control policy can be expressed as

$$u(X(t), M(t), c_D(t), c_W(t)) = \begin{cases} 1 & \text{if } X(t) < S_{c_D,c_W} \text{ and } M(t) = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{2}$$

2.4. Determination of the Optimal Thresholds

The parameters of the marking-dependent control policy are the thresholds that are used based on the markings collected from the system in real time. There are $C_D \times C_W$ parameters of the control policy corresponding to different thresholds $S = \{S_{c_D,c_W}\}$, $c_D \in \{1, 2, ..., C_D\}$, $c_W \in \{1, 2, ..., C_W\}$.

A joint simulation and optimization (JSO) approach is proposed to determine the optimal threshold levels for each pair of production and demand process markings by using the traces gathered from the system [14]. The JSO approach uses a mixed integer programming formulation that captures the discrete-event dynamics of the system when it is controlled with the marking-dependent threshold policy with the specific thresholds, the data collected from the shopfloor, and the simulated inter-event times for other random variables with the available statistical information. This formulation yields the optimal thresholds that minimize the average inventory carrying and backlog cost over a time period. The JSO approach does not impose any assumptions about the demand arrival and production processes.

Alternatively, the demand and production signal arrival processes can be approximated with i.i.d. processes with exponential or phase-type inter-event time distributions or with Markovian Arrival Processes. The parameters of these processes can be determined based on the observed traces. Following the parameter setting, the long-run average cost can be determined analytically for given thresholds and the optimal thresholds can be determined accordingly.

In Section 4, we compare these different methods to determine the optimal thresholds by using the data collected from the lab-scale model of a specific system. We describe the specific production/inventory system analyzed in the experiments in the next subsection.

2.5. Production/Inventory System Model for the Experimental Setup

We build the lab-scale physical model of the system depicted in Figure 2 and used in our experiments. This system has been analyzed analytically and with simulation in [14].

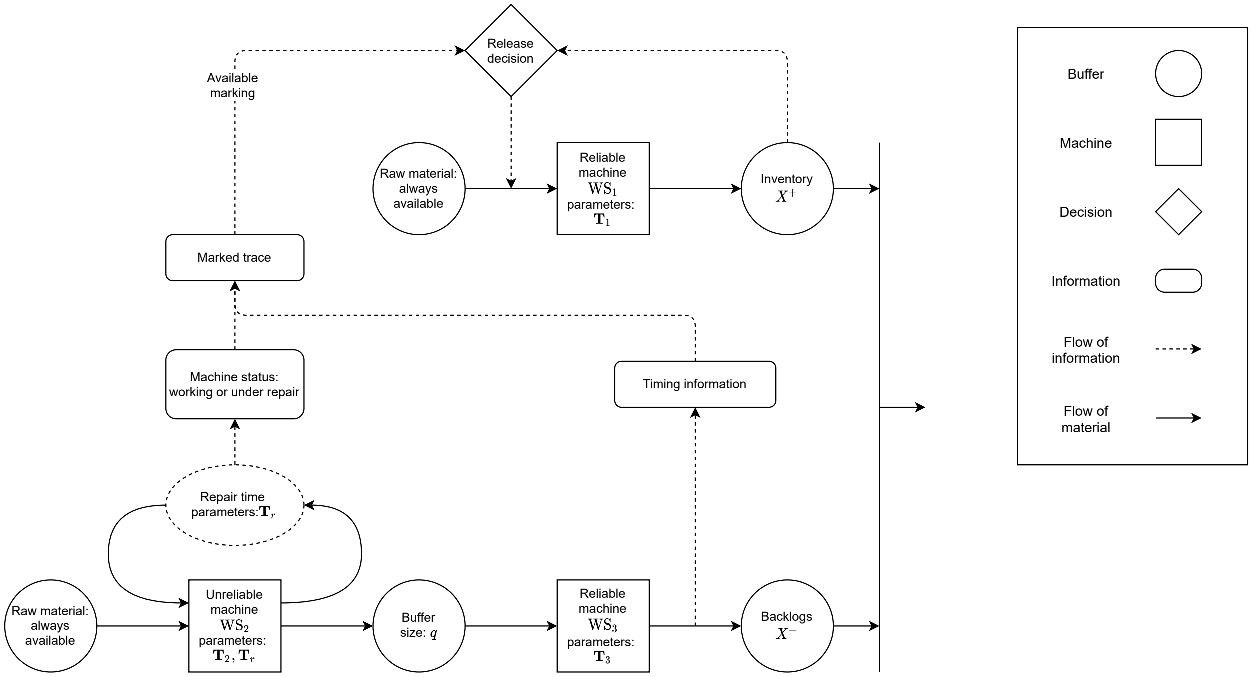


Figure 2: The schematic representation of the LEGO production/inventory system

In this system there are three machines, WS_1, WS_2, WS_3. WS_1, and WS_3 are reliable and WS_2 is unreliable. The production control policy aims at balancing the flow of parts from WS_1 with the flow of parts coming from a two-station production line that is composed of workstations WS_2 and WS_3 separated with a finite buffer by controlling the release of parts to WS_1. The downstream buffer of WS_1 represents the accumulated inventory and the downstream buffer of WS_3 represents the accumulated backlog.

The status of the unreliable workstation WS_2, *in working condition* (1) or *under repair* (2), forms the markings used for making the decision to release parts into WS_1. According to the experimental setup, we do not consider the information about the buffer between WS_2 and WS_3 in forming the markings. Since WS_1 is reliable and has unlimited material, its status is either idle or busy with the processing of an existing part. Since its working status is taken into consideration directly in the authorization decision, we do not consider any additional markings related to production. Therefore, there are two markings coming from the demand process used to authorize the release of parts: $c_D \in \{1, 2\}$. Accordingly, there are two thresholds to control the system: S_1 that is imposed when WS_2 is in working condition and S_2 that is used when WS_2 is down and under repair. Then, the optimal real-time release control policy based on the system status is given as

$$u(X(t), M(t), c_D(t)) = \begin{cases} 1 & \text{if } X(t) < S_{c_D} \text{ and } M(t) = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{3}$$

The long-run average cost of the system when it is operated with this policy is denoted with $\pi(S_1, S_2)$. Let X_2 be the steady-state inventory position of the system when the production is controlled with the double-threshold control policy given in Equation (3), $E[X_2^+]$ be the average inventory level and $E[X_2^-]$ be the average backlog. Then, the average cost $\pi(S_1, S_2)$ for the double-hedging policy can be written as

$$\pi(S_1, S_2) = c^+ E[X_2^+] + c^- E[X_2^-]. \tag{4}$$

As a further approximation, a single threshold S can also be used. In this approximation, the single threshold S, instead of using 2 thresholds, is used to determine the production authorization policy $\tilde{u}(X(t), M(t))$ depending on the system status:

$$\tilde{u}(X(t), M(t)) = \begin{cases} 1 & \text{if } X(t) < S \text{ and } M(t) = 0 \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

The long-run average cost of the system when it is operated with the single-threshold policy is denoted with $\tilde{\pi}(S)$. Let X_1 be the steady-state inventory position of the system when the production is controlled with the single-threshold control policy given in Equation (3), $E[X_1^+]$ be the average inventory level and $E[X_1^-]$ be the average backlog. Then, the average cost for the single-threshold policy is

$$\tilde{\pi}(S) = c^+ E[X_1^+] + c^- E[X_1^-]. \tag{6}$$

The average inventory levels and the backlog levels that are used to determine the average costs in Equation (4) and (6) can be estimated from the data collected from the lab-scale manufacturing system model.

For this system, the system parameters are the processing time parameters for WS_1, WS_2, and WS_3 denoted with the parameter sets $\mathbf{T}_1$, $\mathbf{T}_2$, and $\mathbf{T}_3$, the repair and failure time parameters for WS_2 denoted with $\mathbf{T}_r$, and $\mathbf{T}_f$, and the capacity of the interstation buffer between WS_2, and WS_3 denoted by q. For a given inter-event time process, the parameter set includes the parameters that describes the inter-event time distribution and the autocorrelation function. For example, if the processing times are i.i.d. and exponentially distributed, only the mean of the processing time or the processing rate is included in the parameter set. The system parameters used in the LEGO production/inventory system are given in Table 1.

Table 1: The system parameters

System parameters	Description
$\mathbf{T}_1$	Processing time parameters WS_1
$\mathbf{T}_2$	Processing time parameters of WS_2
$\mathbf{T}_3$	Processing time parameters of WS_3
$\mathbf{T}_r$	Repair time parameters of WS_2
$\mathbf{T}_f$	Failure time parameters of WS_2
q	The capacity of the buffer between WS_2 and WS_3

3. LEGO Production/Inventory System

We have constructed the LEGO production/inventory system as a physical model of the system described in Section 2. Figure 3 depicts the LEGO Production/Inventory system. In Figure 3, the arrows show the direction of the flow of the material in the system.

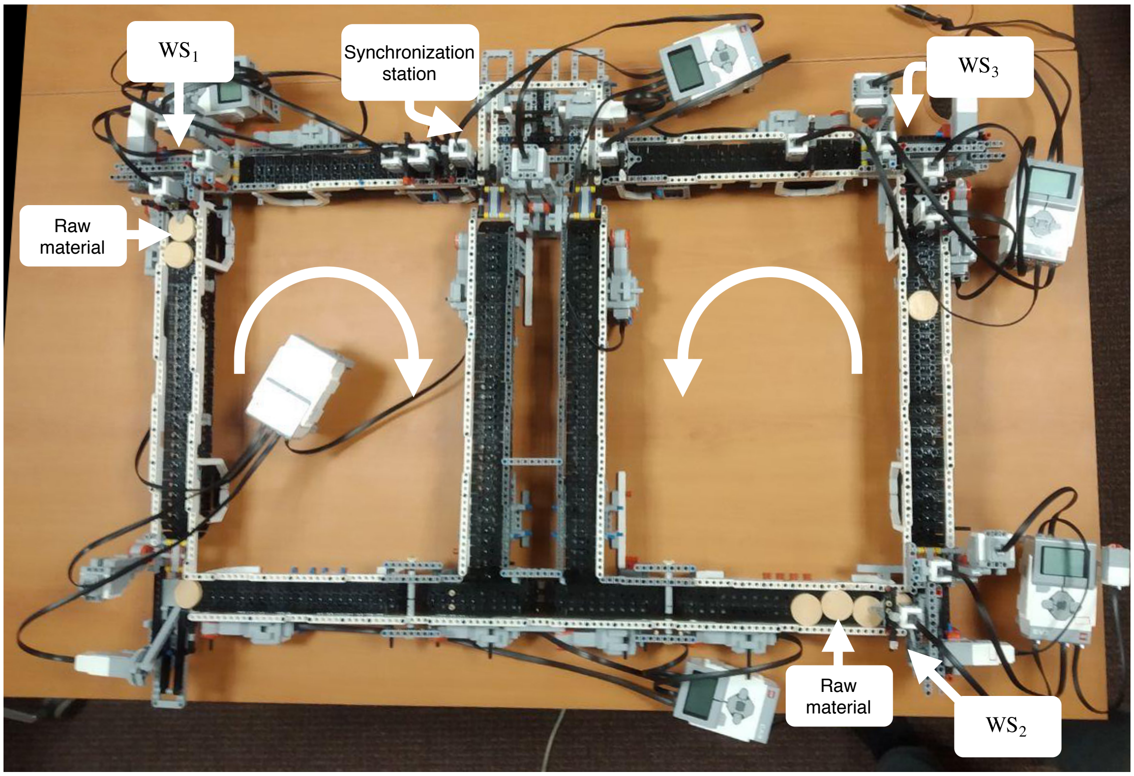


Figure 3: The LEGO production/inventory system

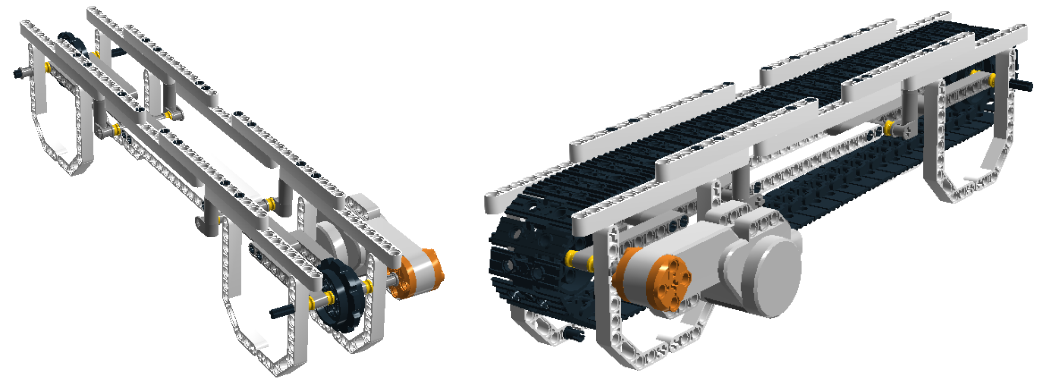
3.1. Construction of System Elements

The lab-scale manufacturing system model is constructed by using EV3 LEGO Mindstorms Education sets. These sets include the LEGO parts for constructing different physical elements such as machines, conveyors, EV3 Intelligent Brick that is a programmable computer to control motors and collect sensor feedback, and different sensors (gyro, ultrasonic, color and touch sensors). We use Matlab to communicate with different EV3 intelligent bricks. The manufacturing system we describe in this section can be built by using 6 EV3 bricks, 11 color sensors and 1 touch sensor.

Here, we describe the construction of the elements of the LEGO production line separately. These elements can be connected together to form different manufacturing systems.

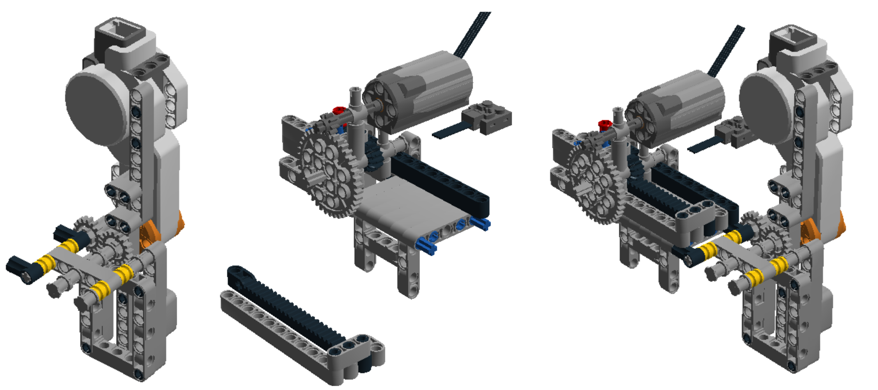
3.1.1. Transportation

Transportation of material between the work stations and the synchronization station is carried out by using conveyors as depicted in Figure 4 (a-b). The conveyors also act as the buffers for the system. The conveyor speed is set to approximately 20 centimeters per second. This speed is set based on its performance in terms of the parts moving through the system correctly. Namely, if the conveyors work slowly, the parts might fail to go through the gates correctly and if the conveyors work very fast, they cause vibrations in the system that at times cause the parts to move out of their correct path and fall out.



(a) Model of a conveyor without the conveyor belt

(b) Model of a conveyor



(c) A work station

(d) A gate mechanism

(e) A gate attached to a work station

Figure 4: The components of the LEGO manufacturing system.

We investigate the effect of the delays caused by the transportation system on the production control policy by changing the *time scale* used to run the lab-scale model with respect to the time scale used in the analytical model in Section 4.

3.1.2. Work Stations and Gates

Work stations are built by using belts that hold the parts while the work station is working and move the part downstream when working on the part is finished. The parts are fed to the workstations using gates. Gates fulfill two purposes. First, they separate the parts from the workstation and allow control of the release of parts into the workstation; second, they provide a force orthogonal to the path of a part allowing for bending the conveyor path. A workstation and a gate are depicted in Figure 4 (c-d) and a gate connected to a workstation is depicted in Figure 4 (e). Figure 5 depicts a workstation and its gate and its upstream and downstream buffers.

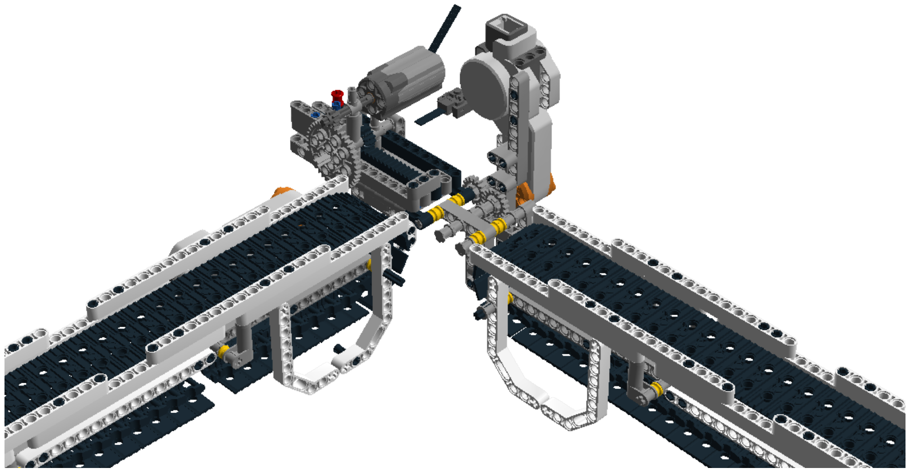
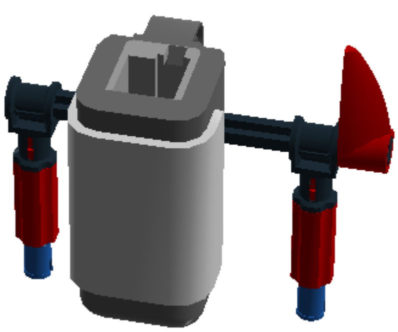


Figure 5: A workstation and its upstream and downstream buffers

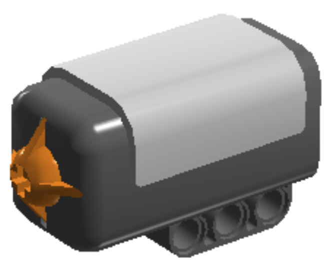
3.1.3. Sensors

Sensors provide the data that will be used by the algorithm that controls the system. We use color sensors that report the color of the object under them. Figure 6 (a) depicts a sensor with its stand that can be installed in different locations in the system for different purposes. However, most of the sensors are used for the control of a workstation subject to blocking and starvation. A workstation subject to blocking and starvation requires three sensors to operate, the *gate sensor* that signals starvation, the *machine sensor* that indicates if a part is in the workstation or not and a *blocking sensor* that signals blocking if the downstream buffer is full. Hence, the placement of the *blocking sensor* for a workstation determines the size of the buffer between that workstation and the downstream work station. Figure 7 depicts the placement of sensors for a workstation subject to blocking and starvation.

A touch sensor is used for starting and stopping the system. A touch sensor has a button that can be pushed for this communicating the start/stop signal. Figure 6 (b) depicts a touch sensor.



(a) A sensor that can be placed in different locations on the conveyors



(b) A touch sensor that is used for starting and stopping the system

Figure 6: The sensors used in the setup

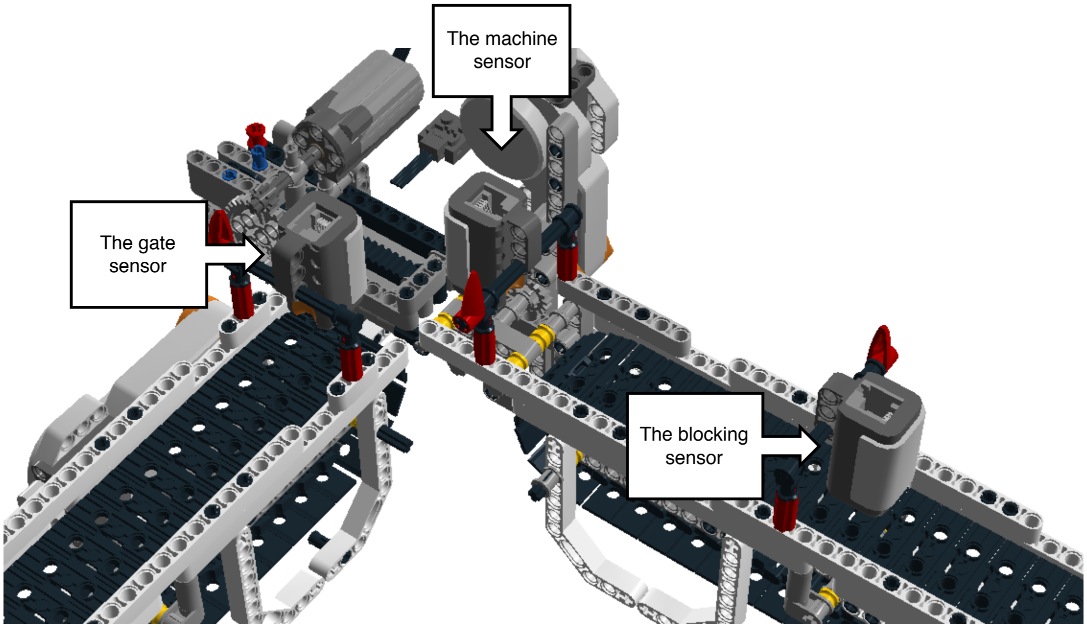
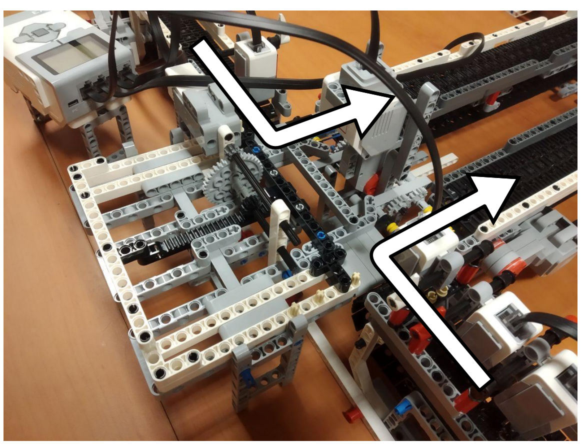


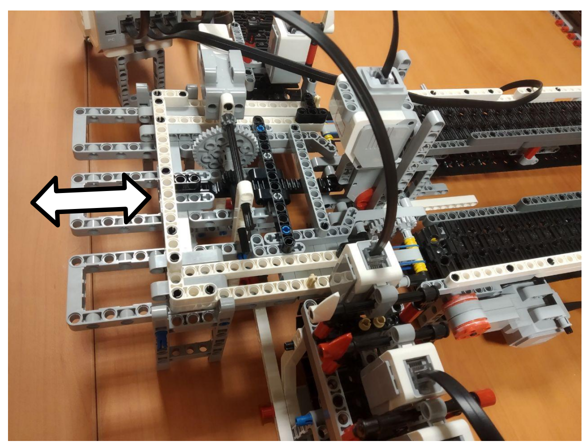
Figure 7: The placement of the sensors for a workstation

3.1.4. Synchronization Station

A synchronization station is a station that allows the material to continue their path only if all the buffers feeding into the synchronization station have at least one item in them. The synchronization station is built using two gate mechanisms and two sensors as depicted in Figure 8. The arrows in Figure 8 (a) show the direction of the flow of the material and the arrow in 8 (b) shows the movement of the gate mechanism for taking the parts in and ejecting them to the downstream buffer.



(a) The path of the parts through the station



(b) Model of a conveyor

Figure 8: The synchronization station

3.1.5. Centralized Control for the LEGO Production/Inventory System

For controlling the system with the marking-dependent policy, the status of WS_2 should be considered to decide on the release of material to WS_1. However as depicted in Figure 3 the sensors related to these workstations are connected to different EV3 bricks. For this reason, the control of the system cannot be totally decentralized. We use a centralized design for the control of the system, where a single algorithm controls all the activities in the system. The system components are connected to EV3 bricks, and the EV3 bricks are connected to the computer that controls the system via bluetooth. A Matlab script on this computer controls the system. In a different setup, an ethernet modem for connecting the EV3 bricks to the computer has been used for controlling the components connected to each EV3 brick and a different Matlab session has been used to control each EV3 brick [27]. Algorithm 1 along with its submodules Algorithms 2-4 given in the Appendix give the pseudocode for controlling the LEGO production/inventory system with the marking-dependent threshold policy.

4. Experiments

The setting we described in Section 3 can be used with different processing time, repair time, and failure time processes that can be set in the program. This setup allows analyzing i.i.d. or correlated interevent times with general distributions. In addition, this setup can be used to investigate different issues related to implementation of various production control policies. For example, performance of static policies, base-stock policies, pull-type policies and other data-driven policies can be investigated with the same setup. The information sources and the information used by the control policies can also be changed. Although, this setup has been developed for research purposes, the same setup can be used for teaching advanced topics in production control, simulation, and manufacturing system design.

In this section, we present the results for a specific setting where the processing times are exponential, the repair time is Erlang, and the failure time is exponential. Furthermore, the information about WS_2 and the number of parts in the finished goods inventory are used as the information utilized by the control policy. The reason for selecting this setup is that the performance of the system under the marking-dependent threshold control policy can be determined analytically. Therefore, the results obtained by using the physical model can be compared directly with the analytical results.

We run two sets of experiments. We first compare the results obtained by using the analytical models and the collected traces from the physical model and discuss the reasons for the disparity. Then, we evaluate performance of the data-driven joint simulation and optimization method on the LEGO production/inventory setup.

4.1. Parameter Setting for the Experiments

The system parameters, the processing rates, the repair rate, failure probability, and the buffer capacity are given in Table 2. In Table 2, the rates are given without the time unit. In order to run the LEGO manufacturing system, the unit of time used for the processing time, repair time, and failure time parameters must be determined. We define the *time scale* parameter, denoted as ϕ to specify the relation between the aforementioned rates and the unit time in the physical system. Namely, the distribution mean in the physical system in seconds is ϕ times the distribution mean given in Table 2 without specifying the time unit. For example, if $\phi = 1$, the average stand-alone processing time of WS_2 will be 1 second and if $\phi = 20$, the average stand-alone processing time of WS_2 will be 20 seconds.

Table 2: The system parameters

System parameters	Distribution	Description	Values
$\mathbf{T}_1 = \{\mu_1\}$	Exponential	Processing rate for WS_1	$\{1\}$
$\mathbf{T}_2 = \{\mu_2\}$	Exponential	Processing rate for WS_2	$\{1\}$
$\mathbf{T}_3 = \{\mu_3\}$	Exponential	Processing rate for WS_3	$\{1\}$
$\mathbf{T}_r = \{r, \lambda\}$	Erlang	Number of phases (r) and the rate (λ) for each phase for the repair time of WS_2	$\{4, 0.5\}$
$\mathbf{T}_f = \{\gamma\}$	Exponential	Failure rate of WS_2	$\{0.09\}$
q	-	The capacity of the buffer between WS_2 and WS_3	$\{1\}$

The analytical models do not account for the delays introduced by transportation and other delays. Consequently, the results obtained by the analytical model do not change with the time scale parameter ϕ. However, since the conveyor moves at a speed of 20 cm/sec., the transportation delay will affect the results obtained by using the physical model depending on the time scale used in the experiments. As ϕ decreases, the effect of external factors, such as transportation delays that are not accounted for in the analytical model will be more pronounced.

We use four values for ϕ, $\phi \in \{1, 4, 10, 20\}$ to investigate the effect of the balance between the delays caused by the physical model and the delays related to the processing times and repair times in our experiments.

4.2. Comparison of the Properties of the Analytical Models and the Collected Traces from the Physical Model

For each value of ϕ, we run the LEGO production/inventory system with different pairs of policy parameters that are the thresholds S_1 and S_2 for the up and down states of WS_2. For each run of the system, we collect the traces of the demand and production times and the sample path of the inventory position to be able to calculate the average cost for a given set of policy parameters and a given value of ϕ. The data used in these experiments were collected from the system that ran for a total duration of 6.5 hours.

4.2.1. Properties of the Analytical Model of the System

In the production/inventory system shown in Figure 2, the demand arrivals are generated by the output of the same two-station production line WS_2 and WS_3 separated with a finite buffer. Since the processing times are exponential, the output process distribution and the autocorrelation can be determined analytically for the given parameters as given in Figure 9.

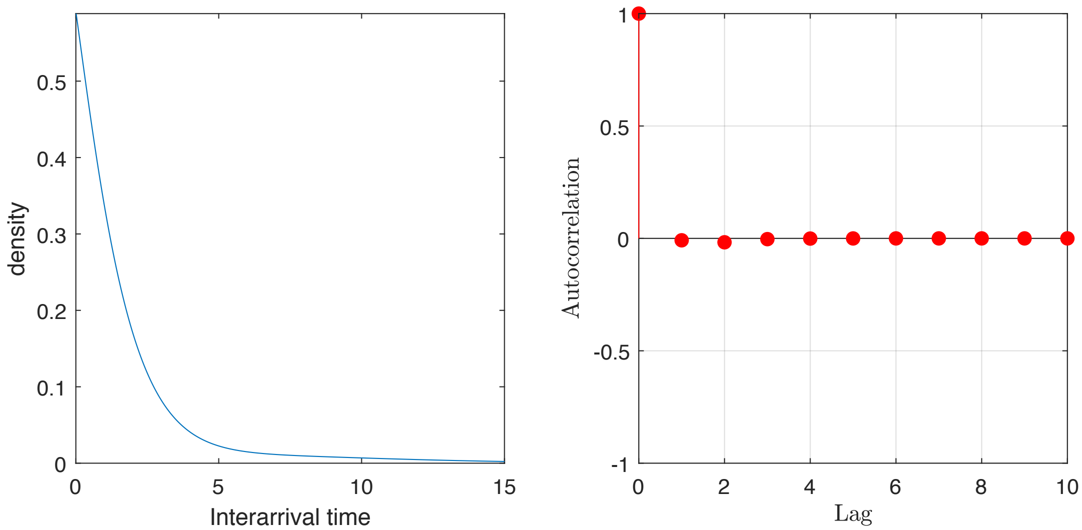


Figure 9: The analytical distribution of the demand inter-arrival times and their autocorrelation function

Using the system parameters given in Table 2, the schematic representation of the system given in Figure 2 and the analytical method given in [14], the system can be evaluated for different values of S_1 and S_2.

Figure 10 depicts the analytical cost function $\pi(S_1, S_2)$ for this set of system parameters and Figure 11 depicts the cost function $\tilde{\pi}(S)$ for a single threshold policy as described in Equations (4)-(6). Figure 10 shows that the analytical models suggest using $S_1 = 3$ and $S_2 = 2$ to minimize the average cost when two thresholds are used for two markings. Furthermore, the cost function is more sensitive to the threshold level when the machine is in working condition. This is because the machine is more often in working condition rather than being repaired. Similarly, Figure 11 shows that using $S = 2$ minimizes the average cost when one threshold is used.

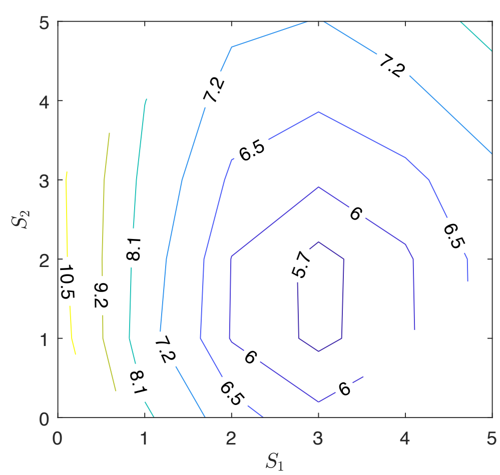


Figure 10: Analytical cost function $\pi(S_1, S_2)$

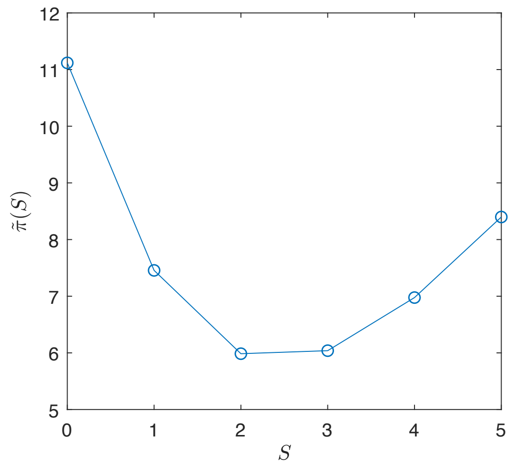


Figure 11: Analytical cost function $\tilde{\pi}(S)$

4.2.2. Properties of the Physical Model

Although, the time scale parameter ϕ does not affect the analytical models, it influences the properties of the data gathered from the physical system because of the delays related to the physical system that are not modeled in the analytical models. As ϕ decreases, the effect of these delays become more important. For very small values of ϕ, the delays mostly determine the inter-event time distributions as opposed to the stand alone processing times.

Figure 12 depicts the inventory position sample paths related to each ϕ value when the production is controlled with the single-threshold policy given in Equation (5). The inventory sample paths are used to determine the average inventory levels and the backlog levels that are used to determine the average costs in Equation (6) and (4). The sample paths are quite different to each other indicating that the production times generated by WS_1 and the demand inter-arrival times generated by WS_2 and WS_3 are different for each ϕ value. As a result, as the time scale parameter ϕ changes, the average inventory and backlog levels and thus the cost functions calculated by using Equations (6) and (4) will be different compared to the analytical cost functions given in Figure 10 and Figure 11. Consequently, the optimal thresholds that minimize the empirical average costs will be different.

Figure 13 depicts the distribution and Figure 14 depicts the autocorrelation of the demand inter-arrival times gathered from the physical system. The empirical distribution and the autocorrelation function depending on the data collected from the physical model are significantly different from the analytical distribution and the autocorrelation function shown in Figure 9. In addition, although the demand arrivals are generated by the output of the same two-station production line WS_2 and WS_3 separated with a finite buffer with the same parameters, the distribution and the autocorrelation function of the inter-departure time from the system are different for different time scale parameters. The analytical model yields the same distribution and the autocorrelation function for the demand inter-arrival times for the given processing rates of WS_2 and WS_3 and the buffer capacity as shown in Figure 9.

Similarly, Figure 15 depicts the processing time distribution of WS_1 based on the traces collected from the system. Although the processing time on WS_1 is generated as an exponentially random variables, the realization on LEGO manufacturing system has a different distribution as a result of the delays related to opening the gates before and after the process completion. These gates are opened with servo motors and introduce a delay that is not accounted in the processing time generation by the software. Similarly, the time scale affects the production time distribution while the analytical model gives the same distribution.

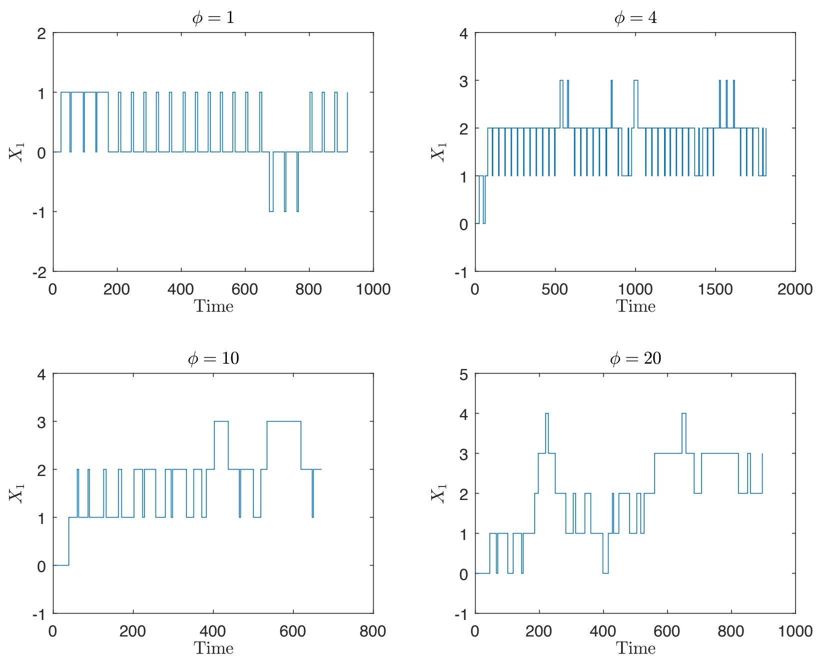


Figure 12: Sample paths of the inventory position gathered from the physical system for different time scale ϕ parameters when the production is controlled with the single-threshold policy

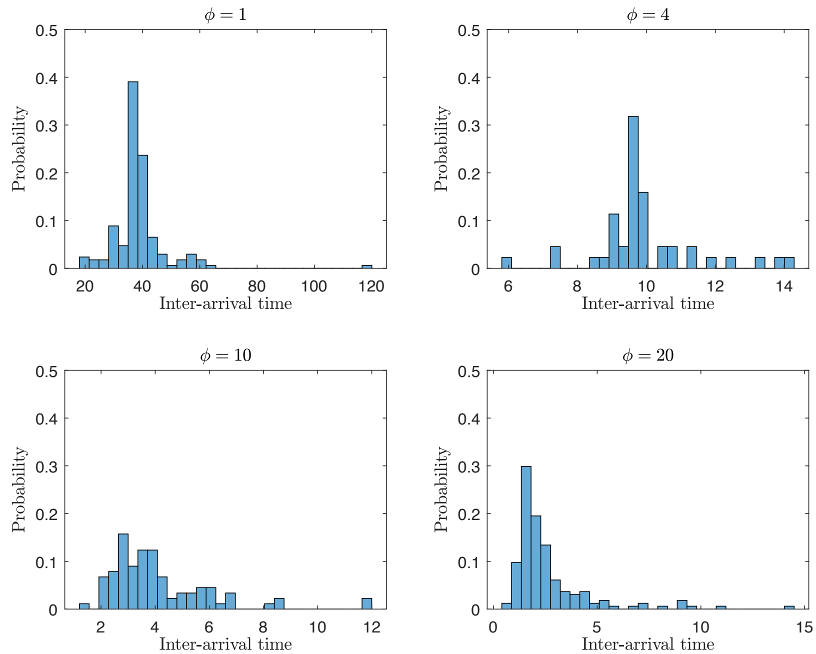


Figure 13: The distribution of the demand inter-arrival times gathered from the physical system for different time scale ϕ when the production is controlled with the single-threshold policy

parameters

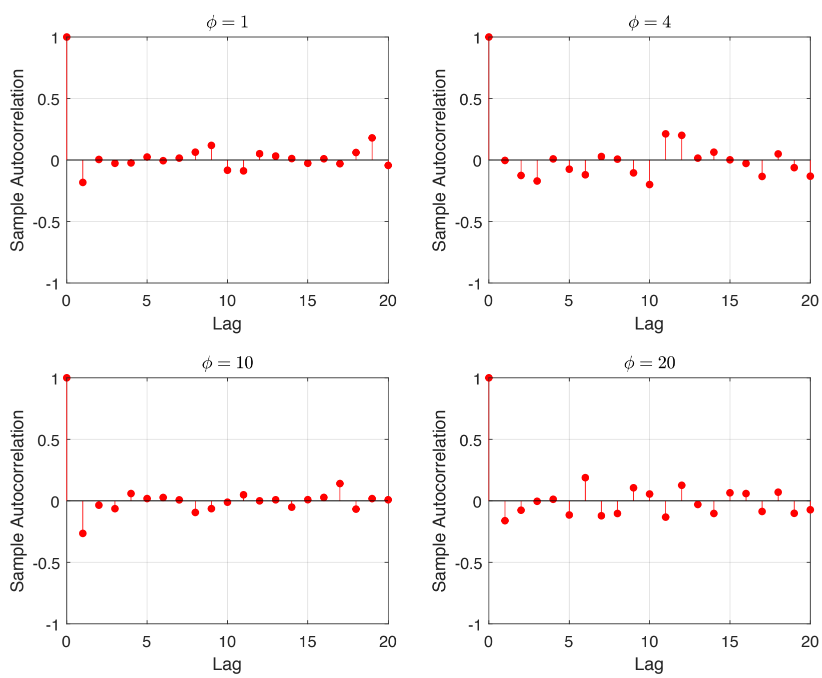


Figure 14: The autocorrelation function of the demand inter-arrival times gathered from the physical system for different time scale ϕ parameters when the production is controlled with the single-threshold policy

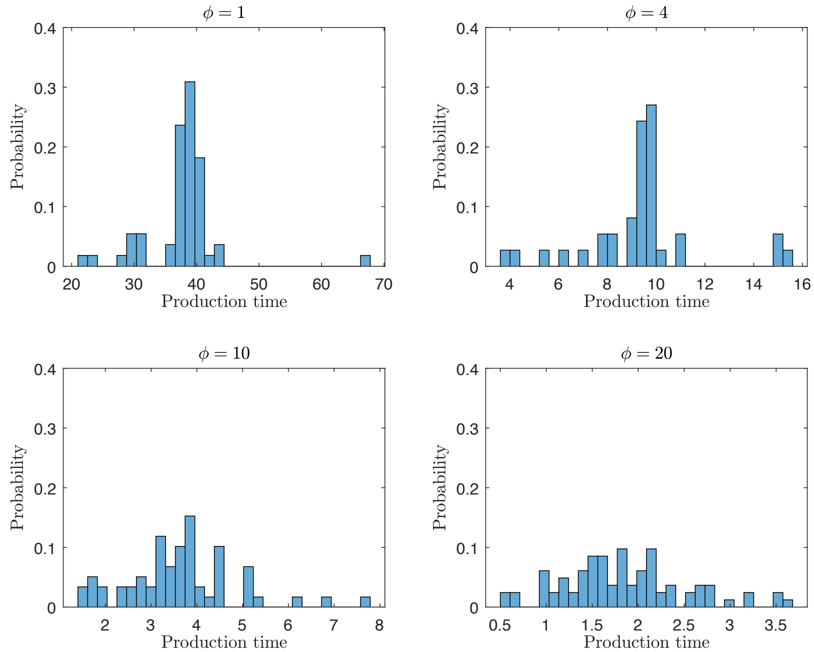


Figure 15: The distribution of the production times gathered from the physical system for different time scale ϕ parameters when the production is controlled with the single-threshold policy

4.2.3. Reasons for the Disparity between the Analytical Models and the Physical Model

The two main reasons that cause a disparity between the analytical models and the empirical data collected from the physical model are ignoring the small delays in the system and correlation between the different inter-event times in the system due to the specifics of the system. In this case, the delays are caused by the conveyors and gates and the dependency is caused by the centralized control of the system. More specifically, for this system, the following factors contribute most to the inaccuracy of the analytical models:

- The travel time of the parts on the conveyors
- The time spent for opening and closing the gates
- The time spent for communications between the physical system and the computer that controls the system
- Time spent for centralized control of the system

In order to develop a more detailed analytical model, the time spent on transportation can be modeled by pseudo-stations with infinite buffers and infinite servers. The time spent during centralized control of the system can also be modeled by using a pseudo station with one machine that has to produce an item for every other workstation in order for them to be able to function. Furthermore, the delay related to opening and closing the gates can be incorporated in the processing times. These delays can also be incorporated in the simulation model.

4.3. Performance of the Data-driven JSO on the LEGO Production/Inventory System setup

In this part, we apply the control methods based on parameter fitting and the data-driven Joint Simulation and Optimization (JSO) method to the traces gathered from the physical system [14]. Here, we consider setting a single threshold for the system. We solve the mixed integer programming formulation with the traces related to demand inter-arrival and processing times collected from the LEGO system and determine the optimal threshold. In addition, we use the collected demand inter-arrival and processing times and fit exponential and phase-type distributions and use the analytical model to determine the optimal threshold. We compare the JSO method with exponential and phase-type distribution fitting approaches, denoted as EXP Fit and PH Fit respectively, to determine the optimal threshold. For the phase-type fitting method where the demand inter-arrival time is approximated with a phase-type distribution, we fit phase-type distributions with up to 10 phases. For all three methods we set the maximum admissible threshold level to 100.

In order to determine the empirical average cost function for the LEGO system, we have used all the available trajectories of the inventory process obtained by running the LEGO manufacturing system with 4 different ϕ values. Based on these trajectories, we built the shortfall processes that indicate the gap between the threshold and the inventory level. Since it is known that the shortfall process does not depend on the threshold level, these shortfall processes are used to estimate the cost for each threshold level and the optimal threshold levels according the empirical average cost function. Figure 16 gives the cost functions generated by this approach. The figure shows that using $S = 3$ is the empirical optimal when the $\phi = 1$ and using $S = 2$ is the empirical optimal when ϕ is equal to 4, 10, and 20.

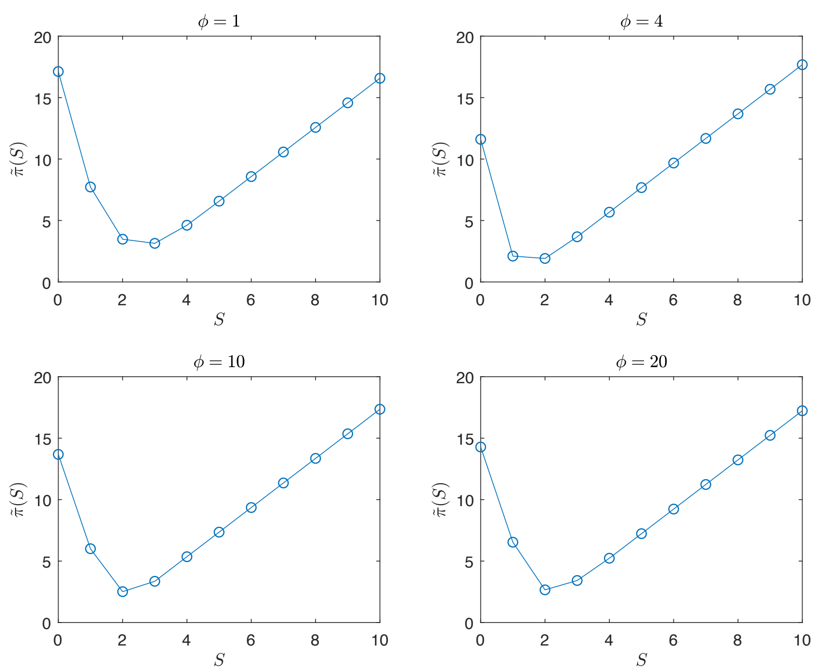


Figure 16: Average cost of the system based on available trajectories of the inventory position

Table 3 gives the thresholds suggested by these three alternative methods and the optimal thresholds that are obtained analytically based on the analytical model with restrictive assumptions and the empirical estimation of the average cost functions given in Figure 16. The optimal analytical threshold for this system is evaluated as 2 as shown in Figure 11.

Table 3: The threshold level suggested by each method

ϕ	Benchmark		Method		
	Analytical	Empirical	JSO	EXP Fit	PH Fit
1	2	3	3	100	64
4	2	2	2	48	6
10	2	2	2	25	4
20	1	2	2	4	2

For small ϕ values, exponential and phase-type fitting do not generate satisfactory answers. This is because the inter-event time distributions for both traces are affected significantly by the delays in the system and do not resemble the exponential distribution when ϕ is relatively small. In other words, there is less randomness in the system, but exponential fitting and phase-type fitting fail to consider this in setting the threshold. This effect is significantly less harmful to JSO. For the largest time scale parameter, phase-type fitting suggests using a threshold closer to the optimal analytical threshold. However, it should be noted that given the disparities between the analytical solutions and the behavior of the system assessing the true optimal solution with the accuracy sufficient to compare the methods needs much longer runs of the physical system with these thresholds.

This experiment shows that the optimal parameters of the marking-dependent threshold policy can be determined effectively by using the joint simulation and optimization approach that uses the real-time data collected from the physical model. As a result, the marking-de

5. Conclusions

In this study, we present a lab-scale manufacturing system model that can be used to investigate different production control policies and analytical and empirical methods to evaluate the performance of manufacturing systems. We demonstrate the construction and the usage of the system by using a LEGO set with sensors, servo motors, programmable computers with a centralized control software developed for Matlab. The proposed setup can be used for research and teaching purposes.

We present two sets of experiments that compare the analytical models with the physical model and evaluate the performance of a data-driven control policy that operates with the traces collected from the physical model. These experiments show that using a physical model yields a more realistic evaluation of a proposed data-driven performance evaluation and production control method as opposed to using the analytical or simulation models that do not include some of the details. These observations can be used to improve the analytical model and the simulation model by incorporating the omitted details that affect the results. In this way, the lab-scale physical models can be used to build a more accurate digital twin that uses analytical models or simulation.

The experiments on the effectiveness of the data-driven control policy used in this study show that the marking-dependent production control policy is considerably less prone to be affected by the differences between the physical system and its analytical representation. This is due to the ability of using real-time data directly in the marking-dependent control policy as opposed to the simplifying assumptions and the parameter fitting steps used in the analytical models. Therefore, using the lab-scale manufacturing system validates the benefits of this data-driven control policy compared to the alternative analytical models.

This work can be extended in different directions. The LEGO manufacturing system can be used for devising and testing data-driven methods that consider the current state of a system and take into account that a good solution must be reached before the system changes drastically. The advantage of using a physical system for evaluating different methods is that the ability of the methods to react to the properties of the system that escape modeling can be assessed. As shown here, even relaxing the common assumptions about independent exponential inter-event times can fall short of matching the analytical models and the behavior of a physical system. To address this, Q-learning can be combined with the data-driven methods to fine-tune the policies in real time. In addition, the data-driven JSO can be used for generating a basis for ordinary behavior of a system for training outlier detection learning methods.

Furthermore, a digital twin that displays the current state of the LEGO manufacturing system and also has the ability to simulate can also be included in the setup. The setup can be revised to use a camera and image processing to track the part movements instead of color sensors used in this setup. A more flexible design that allows changing the information sources used by the control policy can also be developed. These are left for future research.

As a summary, we propose using the lab-scale manufacturing system environment introduced in this paper as a useful tool to investigate, improve, and develop different data-driven control policies before their implementation. Furthermore, we propose the marking-dependent control policy as an effective data-driven production control policy.

References

[1] Elisa Negri, Luca Fumagalli, and Marco Macchi. A review of the roles of digital twin in CPS-based production systems. *Procedia Manufacturing*, 11:939–948, 2017.

[2] Moritz Glatt, Chantal Sinnwell, Li Yi, Sean Donohoe, Bahram Ravani, and Jan C Aurich. Modeling and implementation of a digital twin of material flows based on physics simulation. *Journal of Manufacturing Systems*, 58:231–245, 2021.

[3] Alex Attridge, Mark A Williams, and Charles Tennant. The role of physical modelling in the design verification stage of the automotive NPI process in the premium sector. *International Journal of Automotive Technology and Management*, 7(1):32–54, 2007.

[4] Maria Annunziata Pignataro, Gabriele Lobaccaro, and Giulio Zani. Digital and physical models for the validation of sustainable design strategies. *Automation in Construction*, 39:1–14, 2014.

[5] Kevin Yu, Ashish Kumar Budhiraja, Spencer Buebel, and Pratap Tokekar. Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations. *Journal of Field Robotics*, 36(3):602–616, 2019.

[6] Thomas H-J Uhlemann, Christian Lehmann, and Rolf Steinhilper. The digital twin: Realizing the cyber-physical production system for Industry 4.0. *Procedia CIRP*, 61:335–340, 2017.

[7] Sebastian Haag and Reiner Anderl. Digital twin–proof of concept. *Manufacturing Letters*, 15:64–66, 2018.

[8] Juuso Autiosalo, Riku Ala-Laurinaho, Joel Mattila, Miika Valtonen, Valtteri Peltoranta, and Kari Tammi. Towards integrated digital twins for industrial products: Case study on an overhead crane. *Applied Sciences*, 11(2):683, 2021.

[9] Mengnan Liu, Shuiliang Fang, Huiyue Dong, and Cunzhi Xu. Review of digital twin about concepts, technologies, and industrial applications. *Journal of Manufacturing Systems*, 2020.

[10] Dmitry Ivanov, Christopher S. Tang, Alexandre Dolgui, Daria Battini, and Ajay Das. Researchers' perspectives on Industry 4.0: multi-disciplinary analysis and opportunities for operations management. *International Journal of Production Research*, 59(7):2055–2078, 2021.

[11] Fei Tao, Qinglin Qi, Ang Liu, and Andrew Kusiak. Data-driven smart manufacturing. *Journal of Manufacturing Systems*, 48:157–169, 2018.

[12] Dong-Hee Lee, Jin-Kyung Yang, Cho-Heui Lee, and Kwang-Jae Kim. A data-driven approach to selection of critical process steps in the semiconductor manufacturing process considering missing and imbalanced data. *Journal of Manufacturing Systems*, 52:146–156, 2019.

[13] Lin Li. A systematic-theoretic analysis of data-driven throughput bottleneck detection of production systems. *Journal of Manufacturing Systems*, 47:43–52, 2018.

[14] Siamak Khayyati and Barış Tan. Data-driven control of a production system by using marking-dependent threshold policy. *International Journal of Production Economics*, 226:107607, 2019.

[15] Alp Akcay, Bahar Biller, and Sridhar Tayur. Improved inventory targets in the presence of limited historical demand data. *Manufacturing & Service Operations Management*, 13(3):297–309, 2011.

[16] Zhong-Sheng Hou and Zhuo Wang. From model-based control to data-driven control: Survey, classification and perspective. *Information Sciences*, 235:3–35, 2013.

[17] Matthew J Sobel. The optimality of full service policies. *Operations Research*, 30(4):636–649, 1982.

[18] Panagiotis D Paraschos, Georgios K Koulinas, and Dimitrios E Koulouriotis. Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. *Journal of Manufacturing Systems*, 56:470–483, 2020.

[19] V Polotski, J-P Kenne, and A Gharbi. Kalman filter based production control of a failure-prone single-machine single-product manufacturing system with imprecise demand and inventory information. *Journal of Manufacturing Systems*, 56:558–572, 2020.

[20] Erhan Bayraktar and Michael Ludkovski. Inventory management with partially observed nonstationary demand. *Annals of Operations Research*, 176(1):7–39, 2010.

[21] Kenan Arifoğlu and Süleyman Özekici. Inventory management with random supply and imperfect information: A hidden Markov model. *International Journal of Production Economics*, 134(1):123–137, 2011.

[22] Siamak Khayyati and Barış Tan. A machine learning approach for implementing data-driven production control policies. *International Journal of Production Research*, pages 1–22, 2021.

[23] Young Jae Jang and Vina Sari Yosephine. LEGO robotics based project for industrial engineering education. *The International Journal of Engineering Education*, 32(3):1268–1278, 2016.

[24] Anna Syberfeldt. A LEGO factory for teaching simulation-based production optimization. In *Industrial Simulation Conference, ISC'2010, Budapest, Hungary*, pages 89–94. EUROSIS-ETI, 2010.

[25] Giovanni Lugaresi, Ziwei Lin, Nicla Frigerio, Mengyi Zhang, and Andrea Matta. Active learning experience in simulation class using a LEGO based manufacturing system. In *2019 Winter Simulation Conference*. 2019 Winter Simulation Conference, 2019.

[26] Arturo Sanchez and Jorge Bucio. Improving the teaching of discrete-event control systems using a LEGO manufacturing prototype. *IEEE Transactions on Education*, 55(3):326–331, 2011.

[27] Davide Travaglini. Manufacturing system based on LEGO-robotics: Development of physical and digital models. Master's thesis, Politecnico Milano, Milan, Italy, 2018.

[28] Giovanni Lugaresi, Davide Travaglini, and Andrea Matta. A LEGO manufacturing system as demonstrator for a real-time simulation proof of concept. In *2019 Winter Simulation Conference*. 2019 Winter Simulation Conference, 2019.

[29] Giovanni Lugaresi, Vincenzo Valerio Alba, and Andrea Matta. Lab-scale models of manufacturing systems for testing real-time simulation and production control technologies. *Journal of Manufacturing Systems*, 58:93–108, 2020.

[30] Giovanni Lugaresi and Andrea Matta. Automated manufacturing system discovery and digital twin generation. *Journal of Manufacturing Systems*, 59:51–66, 2021.

[31] Jing-Sheng Song and Paul Zipkin. Inventory control in a fluctuating demand environment. *Operations Research*, 41(2):351–370, 1993.

[32] Rene A Caldentey. *Analyzing the Make-to-Stock queue in the supply chain and eBusiness Settings*. PhD thesis, Massachusetts Institute of Technology, 2001.

Appendix A Pseudo Codes for Controlling the Physical System

Algorithm 1 and its submodules Algorithms 2-4 can be used for controlling the physical system with the marking-dependent threshold policy. The description of the notation used for this algorithm is given in Table 4.

The trace for the processing times $\hat{g}$ is collected directly from the physical system. The demand arrival times $\dot{a}$ are determined based on the the inventory position trajectory X collected from the physical system. That is,

$$\dot{a} = \{a_i : a_i < a_{i+1} \land X(a_i - \epsilon) = X(a_i + \epsilon) - 1\}$$

where ϵ is a small enough number. Then the trace for the demand inter-arrival times $\dot{t}$ can be calculated as

$$\dot{t} = \{a_{i+1} - a_i\}.$$

For the experiment presented in Section 4, Figure 15 depicts the histograms of the processing time traces, $\hat{g}$ collected from the system with different time scales. Similarly, Figures 13 and 14 give the histogram and the autocorrelation of the inter-arrival time traces, $\dot{t}$ collected from the system with different time scales.

Table 4: Description of the variables used in the algorithm for control of the LEGO system.

Variable	Description
T	The value of the clock
α	The status of WS_2
SV	The value for the switch that turns the system on and off
SF_i	Scheduled finishing time for WS_i
$\dot{b}$	A trace of binary indicators that shows if a breakdown will occur or not
$\dot{g}_i$	The trace of processing times for WS_i
$\hat{g}_i$	The trace of processing times for WS_i collected from the system
$\dot{r}_i$	The trace of repair times for WS_2
$i_{g_1}, i_{g_2}, i_{g_3}, i_b, i_r$	Counters for the traces

Algorithm 1 Algorithm for control of the LEGO production/inventory system

```
Initialize EV3s, motors and sensors
Start the clock
X ← 0
i_g1, i_g2, i_g3, i_r, i_b ← 1
SV ← off
α ← in working condition
while T < time limit do
    if The touch sensor is pushed then
        wait for 0.4 seconds
        if The touch sensor is pushed then
            if SV = on then
                Turn the conveyors off
                SV ← off
            else
                Turn the conveyors on
                SV ← on
            end if
        end if
    end if
    if SV = on then
                                                    ▷ WS_2
        Call the submodule for WS_2
                                                    ▷ WS_3
        Call the submodule for WS_3
                                                    ▷ Synchronization Station
        if both of the Synchronization station's sensors detect a part then
            Take a part in from each of the upstream buffers
            Release the parts to the buffer that closes the system's loop
        end if
                                                    ▷ WS_1
        Call the submodule for WS_1
    end if
end while
Save the generated trajectory of X
```

Algorithm 2 Submodule for WS_1

if WS_1's gate sensor sees a part **then**
if WS_1's machine sensor does not see a part **then**
$o_1 \leftarrow \alpha =$ "Under repair" $\wedge$ S_{Down}'s sensor does not see a part
$o_2 \leftarrow \alpha =$ "In working condition" $\wedge$ S_{Up}'s sensor does not see a part
if $o_1 \vee o_2$ **then**
$\hat{g}(i_{g_1}) \leftarrow T$
WS_1 takes a part in from the upstream conveyor
$\text{SF}_1 \leftarrow T + \dot{g}_1(i_{g_1})$
end if
end if
end if
if The SW_1's machine sensor sees a part **then**
if $\text{SF}_1 < T$ **then**
WS_1 releases a part to the downstream conveyor
$X \leftarrow X + 1$
$\hat{g}(i_{g_1}) \leftarrow T - \hat{g}(i_{g_1})$
$i_{g_1} \leftarrow i_{g_1} + 1$
end if
end if

Algorithm 3 Submodule for WS_2

if WS_2's gate sensor sees a part **then**
if WS_2's machine sensor does not see a part **then**
WS_2 takes a part in from the upstream conveyor
if $\dot{b}(i_b)$ **then**
$\text{SF}_2 \leftarrow T + \dot{g}_2(i_{g_2}) + \dot{r}(i_r)$
$i_{g_2} \leftarrow i_{g_2} + 1$
$i_r \leftarrow i_r + 1$
$\alpha \leftarrow$ Under repair
else
$\text{SF}_2 \leftarrow T + \dot{g}_2(i_{g_2})$
$i_{g_2} \leftarrow i_{g_2} + 1$
end if
$i_b \leftarrow i_b + 1$
end if
end if
if The SW_2's machine sensor sees a part **then**
if $\text{SF}_2 < T \wedge \text{SW}_2$'s blocking sensor does not see a part **then**
WS_2 releases a part to the downstream conveyor
$\alpha \leftarrow$ In working condition
end if
end if

Algorithm 4 Submodule for WS_3

if WS_3's gate sensor sees a part **then**
    if WS_3's machine sensor does not see a part **then**
        WS_3 takes a part in from the upstream conveyor
        $\mathrm{SF}_3 \leftarrow T + \dot{g}_3(i_{g_3})$
        $i_{g_3} \leftarrow i_{g_3} + 1$
    end if
end if
if The SW_3's machine sensor sees a part **then**
    if $\mathrm{SF}_3 < T \wedge \mathrm{SW}_3$'s blocking sensor does not see a part **then**
        WS_3 releases a part to the downstream conveyor
        $X \leftarrow X - 1$
    end if
end if